Steady and Unsteady Solutions of the Incompressible Navier-Stokes Equations

Stuart E. Rogers and Dochan Kwak
NASA Ames Research Center, Moffett Field, CA 94035-1000, USA

Cetin Kiris
Stanford University, Stanford, California

Originally appeared as AIAA Paper 89-0463, January 1989
Published in *AIAA Journal.* Vol. 29, No. 4, April 1991, pp. 603–610.

Abstract

An algorithm for the solution of the incompressible Navier-Stokes equations in three-dimensional generalized curvilinear coordinates is presented. The algorithm can be used to compute both steady-state and time-dependent flow problems. The algorithm is based on the method of artificial compressibility and uses a third-order flux-difference splitting technique for the convective terms and a second-order central difference for the viscous terms. Time accuracy is obtained in the numerical solutions by subiterating the equations in pseudo-time for each physical time step. The equations are solved with a line-relaxation scheme which allows the use of very large pseudo-time steps leading to fast convergence for steady-state problems as well as for the subiterations of time-dependent problems. The steady-state solution of flow through a square duct with a $90°$ bend is computed and the results are compared with experimental data. Good agreement is observed. Computations of unsteady flow over a circular cylinder are presented and compared to other experimental and computational results. Finally, the flow through an artificial heart configuration with moving boundaries is calculated and presented.

Introduction

Numerical solutions to the incompressible Navier-Stokes equations are in greater demand than ever before as the field of computational fluid dynamics (CFD) increases its impact as an engineering tool. Problems which can be addressed by the incompressible Navier-Stokes equations include low-speed flows in aerodynamics, internal flows in propulsion, and even problems in biomedical fluid analysis. The more efficient a code can become, the more useful a tool it will be for analysis. Therefore, there is a continuing interest in finding solution methodologies which will produce results using the least amount of computing time. This is particularly true for unsteady, three-dimensional (3-D) problems. Time-accurate solutions of the incompressible Navier-Stokes equations are particularly time consuming because of the elliptical nature of the governing equations. A disturbance at one point in space affects the entire flow domain instantaneously. This requires that the numerical algorithm propagate information through the entire flow domain during one discrete time step. Another important item which directs the development of software is the current trend of new hardware availability. The newest generation of supercomputers has provided an order of magnitude increase in the available processor memory. This can be used to efficiently implement memory intensive algorithms which would otherwise be too costly.

Recent work by the authors[1,2] has lead to the development of a new solution procedure utilizing the method of artificial compressibility and an upwind differencing technique. The current work an extension of the two-dimensional (2-D) flow code into 3-D, as well as further validation and analysis of the method. The artificial compressibility approach has been used successfully by a number of other authors.[3-5] The advantages of using this method are that it directly couples the pressure and velocity fields at the same time level, and produces a hyperbolic system of equations. The artificial waves are merely a mechanism for propagating information throughout the domain, and driving the divergence of velocity toward zero. An appropriate method for applying finite differences to a hyperbolic system is to use the direction of signal propagation to bias the differencing stencil. Hence, some of the upwind differencing schemes which have recently been developed for the compressible Euler and Navier-Stokes equations by a number of authors[6-9] can be utilized. Using the method of Roe[6] the convective terms are differenced by an upwind method that is biased by the signs of the eigenvalues of the local flux Jacobian. This is accomplished by casting the governing equations in their characteristic form and then forming the differencing stencil such that it accounts for the direction of wave propagation.

The time-dependent solution capability comes from extending the artificial compressibility method by subiterating at each physical time step and driving the divergence of velocity toward zero. Introducing the artificial compressibility relation for each physical time-step produces a hyperbolic system of equations; the same mechanism which is used for steady-state solutions is used to obtain the time-dependent solution for each discrete physical time-step. This is in contrast to other methods which utilize primitive variables, many of these require the solution of a poisson equation in pressure at each time step. For example, see Harlow and Welch[10] who developed the MAC method, or Rosenfeld[11], who applied the fractional step method. The artificial compressibility approach requires only that the subiterations be performed until the divergence of velocity has reached some desired order of accuracy. The choice of this order of accuracy in no way effects the stability of the computation. This generally requires only several subiterations per physical time-step. Some of the other methods which solve a Poisson equation in pressure, such as the MAC method of Harlow and Welch[10], must do so to machine accuracy to maintain stability and to avoid accumulating error in the divergence of velocity from one time step to the next.

In the following sections, the details of the artificial compressibility scheme and its use in solving the incompressible Navier-Stokes equations for steady-state and time-dependent problems are given. Details of the implicit line-relaxation procedure are presented. The computed results include the flow through a curved square duct, for which the numerical results are compared with experimental data. The time-accurate capability is tested by computing flow over a circular cylinder, studying both the transient growth of the separation bubble at a Reynolds number of 40, and of the vortex shedding at a Reynolds number of 100. Finally, the unsteady flow through an artificial heart configuration with a moving boundary is presented.

Governing Equations and Artificial Compressibility

The governing equations for time-dependent, incompressible, constant density flow are written in conservative form in generalized coordinates as

$$\begin{aligned} \frac{\partial \hat{u}}{\partial t} &= -\frac{\partial}{\partial \xi}(\hat{e}-\hat{e}_v) - \frac{\partial}{\partial \eta}(\hat{f}-\hat{f}_v) - \frac{\partial}{\partial \zeta}(\hat{g}-\hat{g}_v) = -\hat{r} \\ \frac{\partial}{\partial \xi}\left(\frac{U}{J}\right) &+ \frac{\partial}{\partial \eta}\left(\frac{V}{J}\right) + \frac{\partial}{\partial \zeta}\left(\frac{W}{J}\right) = 0 \end{aligned} \tag{1}$$

For steady-state flow the $\partial/\partial t$ term is zero. In this equation $\hat{r}$ represents the right-hand side of the momentum equations, where J is the Jacobian of the transformation and

$$\hat{u} = \frac{1}{J}\begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\hat{e} = \frac{1}{J}\begin{bmatrix} \xi_x p + uU + \xi_t u \\ \xi_y p + vU + \xi_t v \\ \xi_z p + wU + \xi_t w \end{bmatrix} \quad \hat{f} = \frac{1}{J}\begin{bmatrix} \eta_x p + uV + \eta_t u \\ \eta_y p + vV + \eta_t v \\ \eta_z p + wV + \eta_t w \end{bmatrix} \quad \hat{g} = \frac{1}{J}\begin{bmatrix} \zeta_x p + uW + \zeta_t u \\ \zeta_y p + vW + \zeta_t v \\ \zeta_z p + wW + \zeta_t w \end{bmatrix}$$

$$U = \xi_x u + \xi_y v + \xi_z w$$
$$V = \eta_x u + \eta_y v + \eta_z w$$
$$W = \zeta_x u + \zeta_y v + \zeta_z w$$

In deriving the viscous fluxes, constant viscosity was assumed for simplicity and because initially only laminar flow calculations are being performed. This simplification is not necessary and will be removed in the future. The viscous fluxes are then given by

$$\hat{e}_v = \frac{1}{ReJ}\begin{bmatrix} (\nabla\xi\cdot\nabla\xi)u_\xi & +(\nabla\xi\cdot\nabla\eta)u_\eta & +(\nabla\xi\cdot\nabla\zeta)u_\zeta \\ (\nabla\xi\cdot\nabla\xi)v_\xi & +(\nabla\xi\cdot\nabla\eta)v_\eta & +(\nabla\xi\cdot\nabla\zeta)v_\zeta \\ (\nabla\xi\cdot\nabla\xi)w_\xi & +(\nabla\xi\cdot\nabla\eta)w_\eta & +(\nabla\xi\cdot\nabla\zeta)w_\zeta \end{bmatrix}$$
$$\hat{f}_v = \frac{1}{ReJ}\begin{bmatrix} (\nabla\eta\cdot\nabla\xi)u_\xi & +(\nabla\eta\cdot\nabla\eta)u_\eta & +(\nabla\eta\cdot\nabla\zeta)u_\zeta \\ (\nabla\eta\cdot\nabla\xi)v_\xi & +(\nabla\eta\cdot\nabla\eta)v_\eta & +(\nabla\eta\cdot\nabla\zeta)v_\zeta \\ (\nabla\eta\cdot\nabla\xi)w_\xi & +(\nabla\eta\cdot\nabla\eta)w_\eta & +(\nabla\eta\cdot\nabla\zeta)w_\zeta \end{bmatrix} \qquad (2)$$
$$\hat{g}_v = \frac{1}{ReJ}\begin{bmatrix} (\nabla\zeta\cdot\nabla\xi)u_\xi & +(\nabla\zeta\cdot\nabla\eta)u_\eta & +(\nabla\zeta\cdot\nabla\zeta)u_\zeta \\ (\nabla\zeta\cdot\nabla\xi)v_\xi & +(\nabla\zeta\cdot\nabla\eta)v_\eta & +(\nabla\zeta\cdot\nabla\zeta)v_\zeta \\ (\nabla\zeta\cdot\nabla\xi)w_\xi & +(\nabla\zeta\cdot\nabla\eta)w_\eta & +(\nabla\zeta\cdot\nabla\zeta)w_\zeta \end{bmatrix}$$

where Re is the Reynolds number and where both the velocity gradients in the viscous fluxes and the metrics of the transformation were written as

$$\frac{\partial u}{\partial \xi} = u_\xi, \text{ etc.} \quad \frac{\partial \xi}{\partial x} = \xi_x, \text{ etc.}$$

Steady-state formulation

The artificial compressibility relation is introduced by adding a pseudo-time derivative of pressure to the continuity equation

$$\frac{\partial p}{\partial \tau} = -\beta\left[\frac{\partial}{\partial\xi}\left(\frac{U}{J}\right) + \frac{\partial}{\partial\eta}\left(\frac{V}{J}\right) + \frac{\partial}{\partial\zeta}\left(\frac{W}{J}\right)\right] \qquad (3)$$

where τ is a pseudo-time variable, and is in no way related to the physical time t. Also, a pseudo-time derivative of velocity is added to the momentum equations. In the steady-state formulation the equations are to be marched in pseudo-time until the right-hand side $\hat{r}$ in Eq. (1) and the divergence of velocity converges to zero. For steady-state solutions, derivatives with respect to time t are zero. Thus the resulting system of equations is obtained

$$\frac{\partial \hat{D}}{\partial \tau} = -\frac{\partial}{\partial\xi}(\hat{E} - \hat{E}_v) - \frac{\partial}{\partial\eta}(\hat{F} - \hat{F}_v) - \frac{\partial}{\partial\zeta}(\hat{G} - \hat{G}_v) = -\hat{R} \qquad (4)$$

where $\hat{R}$ is defined here as the residual vector of these equations and where

$$\hat{D} = \frac{1}{J}\begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix}$$

$$\hat{E} = \begin{bmatrix} \beta U/J \\ \hat{e} \end{bmatrix} \quad \hat{F} = \begin{bmatrix} \beta V/J \\ \hat{f} \end{bmatrix} \quad \hat{G} = \begin{bmatrix} \beta W/J \\ \hat{f} \end{bmatrix} \tag{5}$$

$$\hat{E}_v = \begin{bmatrix} 0 \\ \hat{e}_v \end{bmatrix} \quad \hat{F}_v = \begin{bmatrix} 0 \\ \hat{f}_v \end{bmatrix} \quad \hat{G}_v = \begin{bmatrix} 0 \\ \hat{g}_v \end{bmatrix}$$

The pseudo-time derivative is replaced by an implicit Euler finite-difference formula giving

$$\frac{\hat{D}^{m+1} - \hat{D}^m}{\Delta\tau} = -\hat{R}^{m+1}$$

where the superscript m denotes quantities at the mth pseudo-time iteration level. The right-hand side is linearized resulting in

$$\left[\frac{1}{\Delta\tau}I + \left(\frac{\partial\hat{R}}{\partial\hat{D}}\right)^m\right](\hat{D}^{m+1} - \hat{D}^m) = -\hat{R}^m \tag{6}$$

where I is a 4 x 4 identity matrix. This equation is iterated in pseudo-time until the solution converges, at which time the original steady-state incompressible Navier-Stokes equations are satisfied. If Eq. (6) were solved exactly as it is, then for very large $\Delta\tau$ this would become a Newton iteration for a steady-state solution. Not only is it difficult to solve this equation exactly due to the bandwidth of the left-hand side matrix, but also it is not feasible to form the exact Jacobian of the residual vector $\hat{R}$. This is because with the use of upwind differencing, it would require the evaluation of a third-order tensor. Before these details are discussed, however, an equation similar to Eq. (6) for time-dependent problems will be developed.

Time-accurate formulation

In the time-accurate formulation the time derivatives in the momentum equations are differenced using a second-order, three-point, backward-difference formula

$$\frac{3\hat{u}^{n+1} - 4\hat{u}^n + \hat{u}^{n-1}}{2\Delta t} = -\hat{r}^{n+1} \tag{7}$$

where the superscript n denotes the quantities at time $t = n\Delta t$ and $\hat{r}$ is the right-hand side given in Eq. (1). To solve Eq. (7) for a divergence free velocity at the *n+1* time level, a pseudo-time level is introduced and is denoted by a superscript *m*. The equations are iteratively solved such that $\hat{u}^{n+1,m+1}$ approaches the new velocity $\hat{u}^{n+1}$ as the divergence of $\hat{u}^{n+1,m+1}$ approaches zero. To drive the divergence of this velocity to zero, the artificial compressibility relation is introduced in a fashion very similar to the steady-state case.

$$\frac{\partial\hat{D}}{\partial\tau} + \frac{1}{2\Delta\tau}\begin{bmatrix} 0 \\ 3\hat{u}^{n+1} - 4\hat{u}^n + \hat{u}^{n-1} \end{bmatrix} = -\hat{R}^{n+1} \tag{8}$$

where $\hat{D}$ is the same vector defined in Eq. (5) and $\hat{R}$ is the same residual vector defined in Eq. (4). When this equation is iterated in pseudo time until $\partial\hat{D}/\partial\tau = 0$, Eq. (7) is satisfied and the divergence of velocity is zero. Applying an implicit Euler finite difference to the pseudo-time derivative gives

$$\begin{aligned} & I_{t\tau}(\hat{D}^{n+1,m+1} - \hat{D}^{n+1,m}) \\ & = -\hat{R}^{n+1,m+1} - \frac{I_m}{\Delta t}(1.5\hat{D}^{n+1,m} - 2\hat{D}^{n} + 0.5\hat{D}^{n-1}) \end{aligned} \tag{9}$$

where

$$\begin{aligned} I_{t\tau} &= \text{diag}\left[\frac{1}{\Delta\tau},\ \frac{1}{\Delta\tau} + \frac{1.5}{\Delta t},\ \frac{1}{\Delta\tau} + \frac{1.5}{\Delta t},\ \frac{1}{\Delta\tau} + \frac{1.5}{\Delta t}\right] \\ I_m &= \text{diag}[0,\ 1,\ 1,\ 1] \end{aligned}$$

Finally, the residual term at the *m+1* pseudo-time level is linearized giving the following equation in delta form

$$\begin{aligned} & \left[I_{t\tau} + \left(\frac{\partial\hat{R}}{\partial\hat{D}}\right)^{n+1,m}\right](\hat{D}^{n+1,m+1} - \hat{D}^{n+1,m}) \\ & = -\hat{R}^{n+1,m} - \frac{I_m}{\Delta t}(1.5\hat{D}^{n+1,m} - 2\hat{D}^{n} + 0.5\hat{D}^{n-1}) \end{aligned} \tag{10}$$

Note that there is no linearization done in physical time. Equation (10) is very similar to the steady-state formulation given by Eq. (6). In a sense the time-accurate formulation requires the solution of a steady-state problem in order to advance one physical time step. Both systems of equations will require the discretization of the same residual vector $\hat{R}$. The derivatives of the viscous fluxes in this vector are approximated using second-order central differences. The formation of the convective fluxes, however, is not such a simple matter. For this, upwind differencing based on the method of Roe[6] is used. Details of this formulation for the incompressible Navier-Stokes equations can be found in Rogers and Kwak[1]. For all of the results reported in this paper, third-order upwind differencing was used at the interior points, and second-order upwind differencing was used at points next to the boundaries. This upwind scheme make use of the eigensystem of the Jacobian of the convective flux vectors, which is included here, as the results presented in ref.[1] is only for two dimensions. For the current formulation a generalized flux vector is given by

$$\hat{E}_i = \frac{1}{J}\begin{bmatrix} \beta Q \\ k_t u + k_x p + uQ \\ k_t v + k_y p + vQ \\ k_t w + k_z p + wQ \end{bmatrix} \tag{11}$$

where $\hat{E}_i = \hat{E}, \hat{F}, \hat{G}$ for $i = 1, 2,$ and 3 respectively, and the metrics are represented with

$$k_x = \frac{\partial\xi_i}{\partial x}, i = 1,2,3 \quad k_y = \frac{\partial\xi_i}{\partial y}, i = 1,2,3$$

$$k_z = \frac{\partial\xi_i}{\partial z}, i = 1,2,3 \quad k_t = \frac{\partial\xi_i}{\partial t}, i = 1,2,3$$

and the contravariant velocity is

$$Q = k_x u + k_y v + k_z w$$

The Jacobian matrix $A_i = \frac{\partial \hat{E}_i}{\partial \hat{D}}$ of the flux vector in Eq. (11) is given by

$$\hat{A}_i = \begin{bmatrix} 0 & k_x\beta & k_y\beta & k_z\beta \\ k_x & k_x u + Q + k_t & k_y u & k_z u \\ k_y & k_x v & k_y v + Q + k_t & k_z v \\ k_z & k_x w & k_y w & k_z w + Q + k_t \end{bmatrix} \tag{12}$$

A similarity transform for the Jacobian matrix is introduced $\hat{A}_i = X_i \Lambda_i X_i^{-1}$ where

$$\begin{gathered} \Lambda_i = \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3, \lambda_4] \\ \lambda_1 = Q + k_t \quad \lambda_2 = Q + k_t \\ \lambda_3 = Q + k_t/2 + c \quad \lambda_4 = Q + k_t/2 - c \end{gathered} \tag{13}$$

and where c is the scaled artificial speed of sound given by

$$c = \sqrt{(Q + k_t/2)^2 + \beta(k_x^2 + k_y^2 + k_z^2)} \tag{14}$$

The matrix of the right eigenvectors is given by

$$X_i = \begin{bmatrix} 0 & 0 & \beta(c - k_t/2) & -\beta(c + k_t/2) \\ x_k & x_{kk} & u\lambda_3 + \beta k_x & u\lambda_4 + \beta k_x \\ y_k & y_{kk} & v\lambda_3 + \beta k_y & v\lambda_4 + \beta k_y \\ z_k & z_{kk} & w\lambda_3 + \beta k_z & w\lambda_4 + \beta k_z \end{bmatrix} \tag{15}$$

and its inverse is given by

$$X_i^{-1} = \frac{1}{c^2 - k_t^2/4} \begin{bmatrix} x_{kk}(k_z v - k_y w) + y_{kk}(k_x w - k_z u) + z_{kk}(k_y u - k_x v) & y_{kk}(\lambda_1 w + \beta k_z) - z_{kk}(\lambda_1 v + \beta k_y) & z_{kk}(\lambda_1 u + \beta k_x) - x_{kk}(\lambda_1 w + \beta k_z) & x_{kk}(\lambda_1 v + \beta k_y) - y_{kk}(\lambda_1 u + \beta k_x) \\ x_k(k_y w - k_z v) + y_k(k_z u - k_x w) + z_k(k_x v - k_y u) & -y_k(\lambda_1 w + \beta k_z) + z_k(\lambda v + \beta k_y) & -z_k(\lambda_1 u + \beta k_x) + x_k(\lambda_1 w + \beta k_z) & -x_k(\lambda_1 v + \beta k_y) + y_k(\lambda_1 u + \beta k_x) \\ -\lambda_4(c + k_t/2)/(2\beta c) & k_x(c + k_t/2)/(2c) & k_y(c + k_t/2)/(2c) & k_z(c + k_t/2)/(2c) \\ -\lambda_3(c - k_t/2)/(2\beta c) & k_x(c - k_t/2)/(2c) & k_y(c - k_t/2)/(2c) & k_z(c - k_t/2)/(2c) \end{bmatrix} \tag{16}$$

where

$$x_k = \frac{\partial x}{\partial \xi_{i+1}} \quad x_{kk} = \frac{\partial x}{\partial \xi_{i+2}}$$

$$\xi_{i+1} = \eta, \zeta, \text{ or } \xi \text{ for } i = 1, 2, \text{ and } 3 \text{ respectively}$$

$$\xi_{i+2} = \zeta, \xi, \text{ or } \eta \text{ for } i = 1, 2, \text{ and } 3 \text{ respectively}$$

Implicit Scheme

The Eqs. (6) and (10) are numerically represented and solved using a Gauss-Seidal line-relaxation scheme similar to the one used by MacCormack[12] and Chakravarthy.[13] Approximate Jacobians of the residual vector resulting from first-order upwind-differenced fluxes are substituted for the $\frac{\partial R}{\partial D}$ term on the left-hand side. Instead of factoring this banded matrix, it is approximately solved using a line-relaxation. Using this, a sweep direction is chosen such that all terms on the left-hand side from points along this direction are solved for implicitly. All terms on the left-hand side from points off this sweep line are multiplied by the latest known $\Delta\hat{D}$ and moved to the right-hand side, where $\Delta\hat{D}$ is equal to $\hat{D}^{m+1} - \hat{D}^{m}$ for Eq. (6) or $\hat{D}^{n+1,m+1} - \hat{D}^{n+1,m}$ for Eq. (10). The resulting set of equations is a tridiagonal system of 4 x 4 blocks. This system is solved for each line as the domain is swept several times. For the computed results presented in this paper two sweeps are used in each of the coordinate directions, once forward and once backward.

This sweeping process can be efficiently implemented on a vector processor using the following ordering. When, for example, the sweep is chosen such that all points in the i-direction (on a j=constant, k=constant line) are implicit, a forward sweep is carried out by solving each of these i-lines for j=1, k=1 to kmax, then j=2, k=1 to kmax, then j=3, k=1 to kmax, etc. Solving only one block tridiagonal system at a time is inherently recursive and very little vectorization takes place. It is important to perform the sweeps like this such that the most recently computed $\Delta\hat{D}$ is available when forming the right-hand side of the next i-line. In other words the right-hand side terms in a forward sweep will use $\Delta\hat{D}$ at (i,j-1,k) and (i,j,k-1) from the current sweep level, and at (i,j+1,k) and (i,j,k+1) from the previous sweep level. Trying to solve i-lines for j=1 to jmax simultaneously or k=1 to kmax simultaneously will necessarily mean using more $\Delta\hat{D}$ terms from the previous sweep level, and this will severely reduce the convergence. This process can be vectorized in only one way without reducing the use of the current sweep level information, and that is to solve simultaneously for i-lines for whom j+k=constant. This means solving for i-lines which all line on a diagonal plane cutting through the computational domain parallel to the i-direction. The vector length for this process will be rather short at the beginning and ends of the sweeps, but it's implementation can reduce the time spent in the block-tridiagonal solver by a factor of 4. This use of diagonal planes is similar to that used by Yoon et al.[14] in an LU-SGS scheme.

Boundary Conditions

Implicit boundary conditions are used at all of the boundaries which helps make possible the use of large time steps. At a viscous no-slip surface, the velocity is specified to be zero, and the pressure at the boundary is obtained by specifying that the pressure gradient normal to the wall be zero. The boundary conditions used for inflow and outflow regions are based on the method of characteristics. The formulation of these boundary conditions is similar to that given by Merkle and Tsai[15], the details of the current implementation can be found in Rogers and Kwak[1].

Computed Results

Presented here are the results of three different laminar flow computations. These are the flow through a square duct with a 90° bend, the flow over a circular cylinder, and the flow through an artificial heart. For each of the test cases presented, the larger the pseudo-time step $\Delta\tau$ was, the better the convergence was. In all of the cases presented here the solution remained stable no matter how large a pseudo-time step was used, so it was set to 10^{12} which effectively reduced the $1/\Delta\tau$ terms to zero. The choice of β for each case was arrived at through numerical convergence tests. It was found that the convergence was sensitive to the value β, and so its choice is quite important. However, the code will converge well for a large range of β and will become unstable only if extreme values are used. In the range of β for which the code will converge, there is usually a value which will give optimum performance. For steady-state cases this is found to be for β on the order of one to ten, and for time-dependent calculations it is on the order of 10^2 to 10^3.

As reported by the authors in a previous application of the artificial compressibility method[16,17], the parameter β controls the speed of the propagation of the artificial pressure waves. If β is too small, it will not propagate the pressure wave fast enough to keep up with the developing boundary layer and the diffusion of viscosity. These previous works used approximate factorization. The error introduced by approximate factorization included terms of β^2 and β^3, thus placing an upper bound on the allowable value of β. No such upper bound exists in the current application. This is illustrated by example in some of the computations in this next section.

Square Duct with 90° Bend

The flow through a square duct with a 90° bend was used as a steady-state test case. This particular geometry was studied experimentally by Humphrey *et al.*[18] Three different grids were used whose dimensions are 31 x 11 x 11, 61 x 21 x 21, and 121 x 41 x 41. The problem was non-dimensionalized using the side of the square cross-section as the unit length, and the average inflow velocity as the unit velocity. The Reynolds number was 790 based on the unit length and velocity. The grid for the 31 x 11 x 11 case is shown in Fig. 1. The straight inflow section before the bend was set to a length of five and the outflow section downstream of the bend was also set to a length of five. The solutions have been tested and found to be independent of downstream boundary locations and the downstream boundary conditions. The radius of curvature of the inner wall in the curved section was 1.8 units in length. The inflow velocity profile was prescribed to be that of a fully developed laminar straight square duct as given by White.[19]

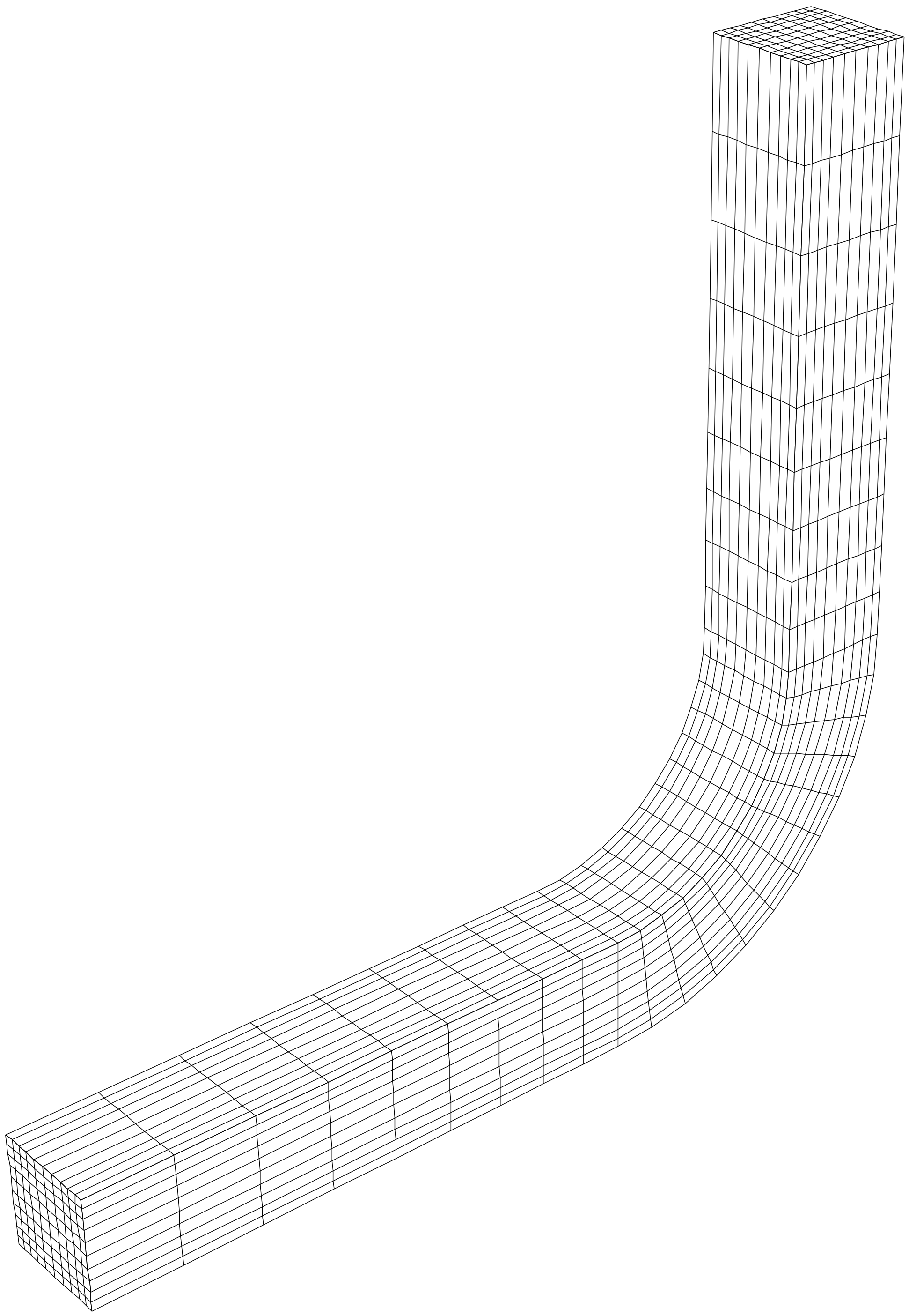


Fig. 1 Geometry and grid (31 x 11 x 11) for computation of flow through a square duct with a 90 degree bend.

As an example of how the code behaves for different values of β, the 31 x 11 x 11 grid was run with β set to values ranging from 0.1 to 10000. The convergence of the maximum residual versus time is plotted in Fig. 2a. This shows that values of β ranging from 1.0 to 100 lead to excellent convergence, and that the code will converge for very large values of β. The remaining computations for this problem used β values of 10.

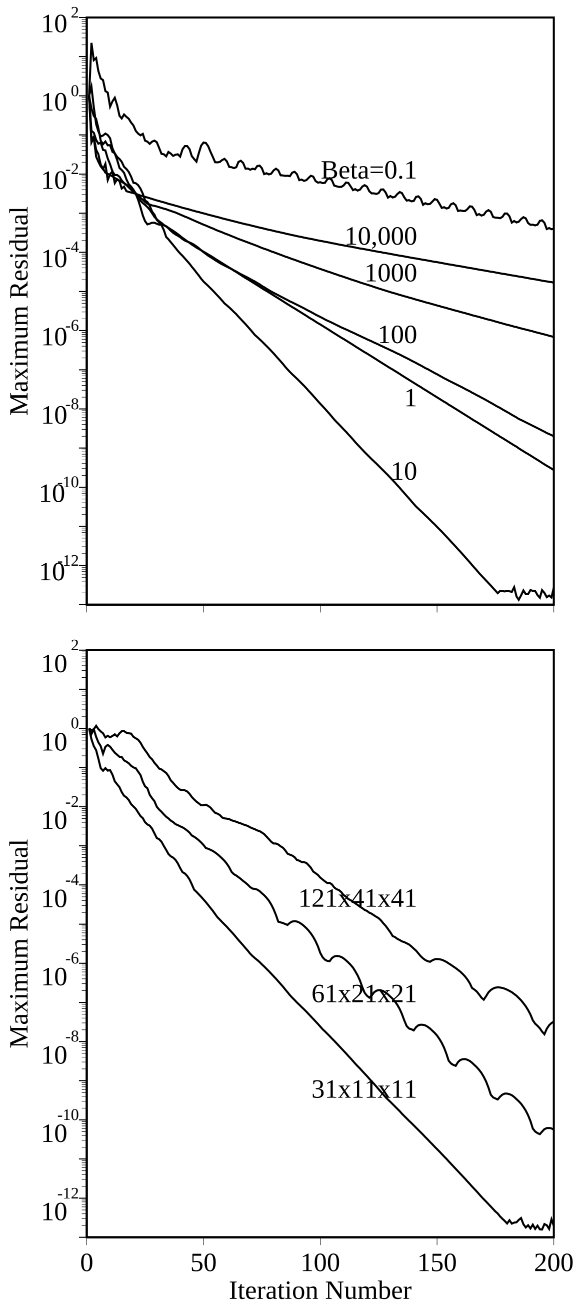


Fig. 2 Maximum residual versus iteration number for the flow through a square duct with a 90 degree bend. a) Using different values of β on 31x11x11 grid. b) Using different grids with $\beta = 10$.

The convergence history for the three grid cases is compared in Fig. 2b. The maximum residual over all the grid points is plotted versus iteration number. The convergence is shown to be very fast, and although it is slower for the finest grid, it is not dramatically so. The slower convergence is to be expected for the finer grids because information has to propagate through a greater distance in computational space. The velocity magnitude contours at the 90 degree cross section at the end of the bend are compared for the 61x21x21 grid (right side) and the 121x41x41 grid (left side) in Fig. 3. This figure shows that there is very good comparison between the medium and the fine grid solutions throughout most of this cross section. In particular the location and value of the maximum velocity magnitude agrees very well. The only difference is that the swirling of the region of flow with higher velocity magnitude near the inner wall is more dissipated in the medium grid.

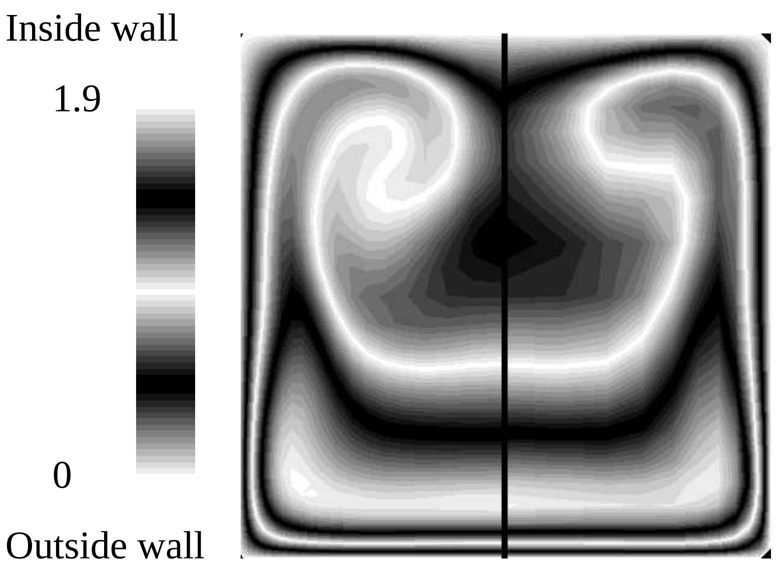


Fig. 3 Velocity magnitude contours for fine grid and medium grid computations at $\theta = 90$ degrees.

The computed results are compared to the experimental results of Humphrey *et al.*[18] in Fig. 4. Shown are the longitudinal velocity profiles at various streamwise stations for two different cross-flow locations. The plots on the right side are from z=0.25, that is half way between the x-y plane wall and the x-y symmetry plane. The second location, shown on the right side from the x-y plane at z=0.5, or the x-y symmetry plane. In each of these figures, the profiles are shown at four positions in the curved section corresponding to θ equal to 0, 30, 60, and 90.° The symbols represent the experimental results and the lines represent the computed fine-grid solution. The results are fairly close to one another except for one trend. The forming of the second maximum in the velocity on the inner wall side occurs further upstream in the computations than in the experiment. This causes the discrepancy at the $\theta = 60$ degree location in both plots, and the $\theta = 90$ degree location for the z=0.25 plot, where three different maxima occur in the computations. More understanding of this can be gained by looking at the entire cross section. In left-hand side of Fig. 5 the velocity magnitude contours from the fine-grid computations are plotted at the $\theta = 30$, 60, and 90 degree cross sections. These show first how the high-velocity fluid moves very close to the outside wall, and it also shows the formation of the swirl which brings some of this high-velocity fluid toward the inner wall, forming the second maximum in velocity near the inside wall. Then at the 90 degree location the swirl has wrapped the region of higher-velocity fluid around toward the middle, which is the mechanism which causes the third maximum in velocity as seen in Fig. 4. It is thought that any small change in the Reynolds number, or even a small amount of turbulent mixing could change this swirling mechanism enough to explain the difference between the computation and the experimental measurements.

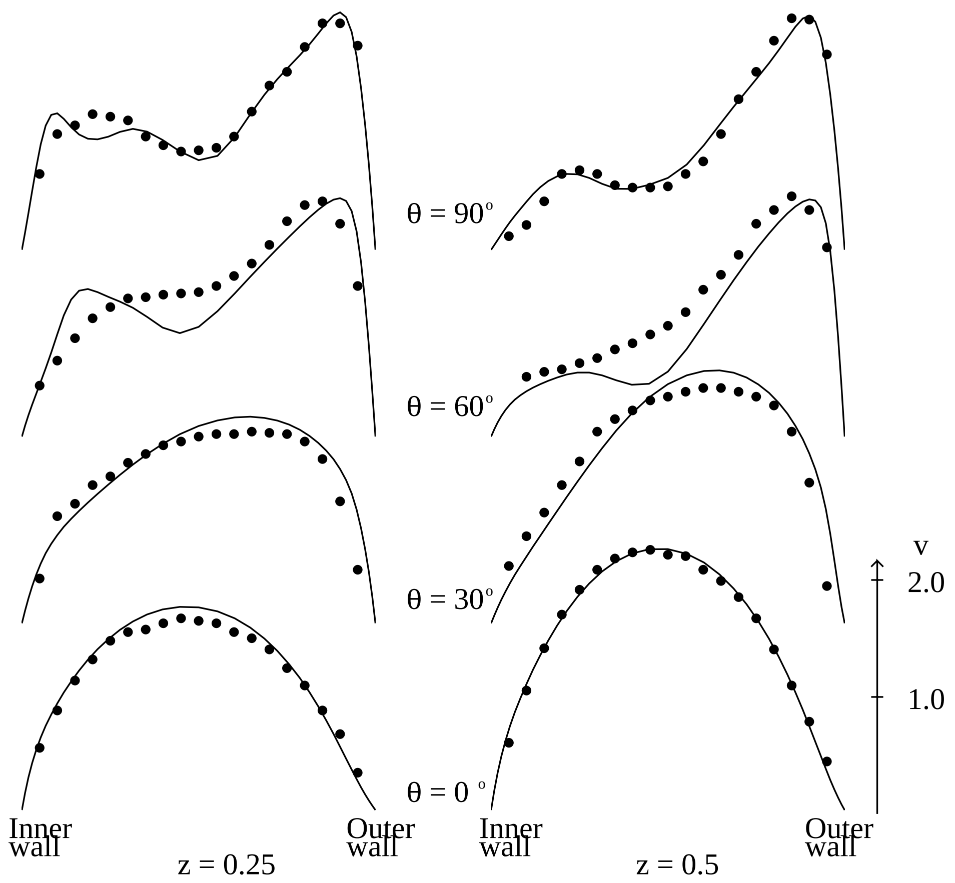


Fig. 4 Computational (solid line) and experimental (dot) streamwise-velocity profiles at four streamwise stations.

Also shown in Fig. 5 are the cross-stream velocity vectors. This figures shows one of the pair of secondary vortices are generated by the large values of static pressure on the outside wall which arises when the flow starts to negotiate the sharp bend. The center of these vortices is seen to move towards the inside wall between the $\theta = 30°$ station and the $\theta = 60°$ position. The vortices tend to center again further downstream, and at the same time a secondary pair of vortices are seen to appear. This agrees qualitatively with the observations of the experiment of Humphrey *et al.*[18]

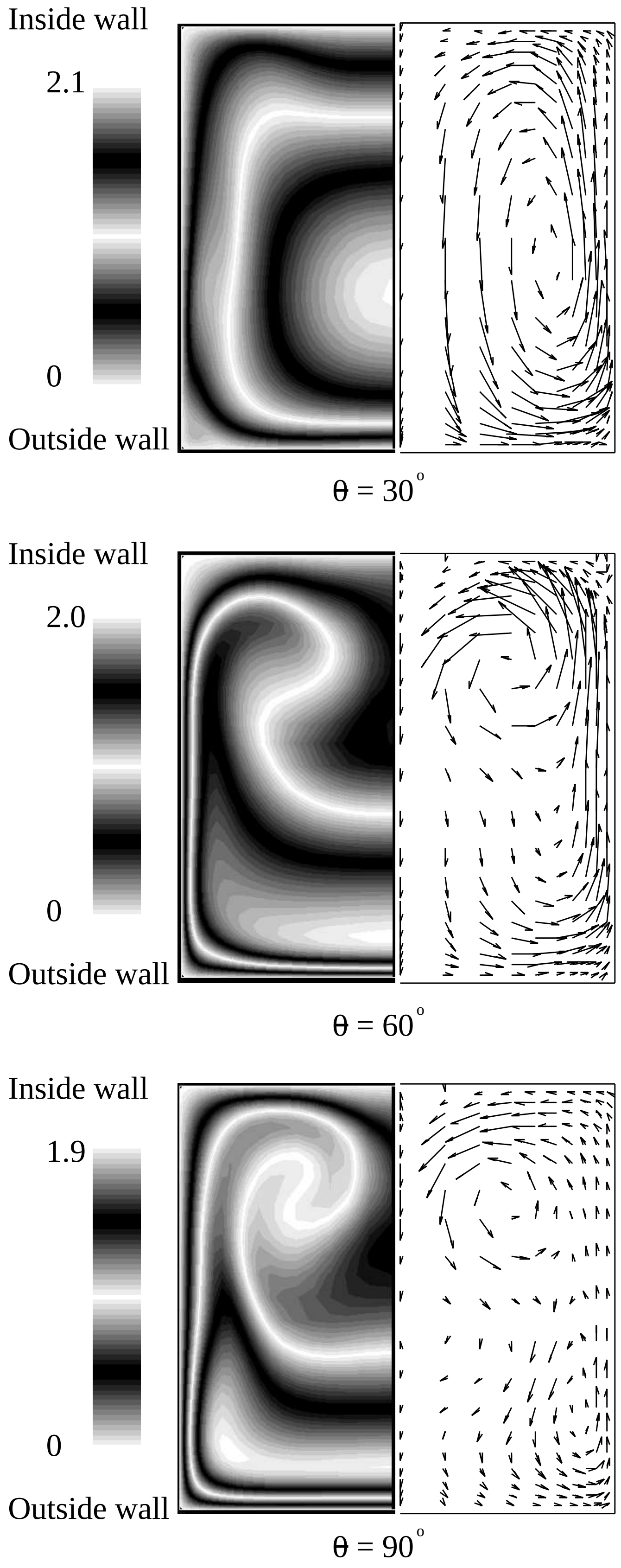


Fig. 5 Velocity magnitude contours and cross-sectional velocities for the curved square duct at three streamwise stations.

Unsteady Flow Over a Circular Cylinder

The accuracy of the unsteady flow computations are verified for the problem of flow over a circular cylinder in 2-D. A 91 x 90 O-grid was used whose spacing at the wall was 0.005 diameters, and whose points were clustered toward the cylinder surface and toward the wake. The problems were run with a β of 100, using line-relaxation sweeps both forward and backward in both coordinate directions. The subiterations were carried out until the maximum value of the divergence of velocity was less than a convergence parameter ϵ_c. These cases were run for different values of ϵ_c ranging from 10^{-1} to $10.^{-4}$

The first problem run with this geometry was the case of an impulsively started flow at a Reynolds number of 40. The separation length behind the cylinder was measured versus time. These results are plotted in Fig. 6 for the first three values of ϵ_c. The results for $\epsilon_c = 10^{-4}$ lie virtually on top of the 10^{-3} results. Also plotted are the computations of Rosenfeld,[11] the experimental values of Coutanceau and Bouard,[20] and the computations of Collins and Dennis.[21] The plot shows that values of ϵ_c of 0.1 does not give satisfactory results, but values lower than that show results quite close to each other. The value of 0.1 requires only two subiterations per time step, the value of 0.01 averaged three subiterations, the value 10^{-3} averaged five subiterations, and the case of 10^{-4} required an average of eight subiterations per time step. Thus the computational time nearly doubled for each order of magnitude that the divergence of velocity was decreased. However, any work done beyond five subiterations per time step did not change the solution.

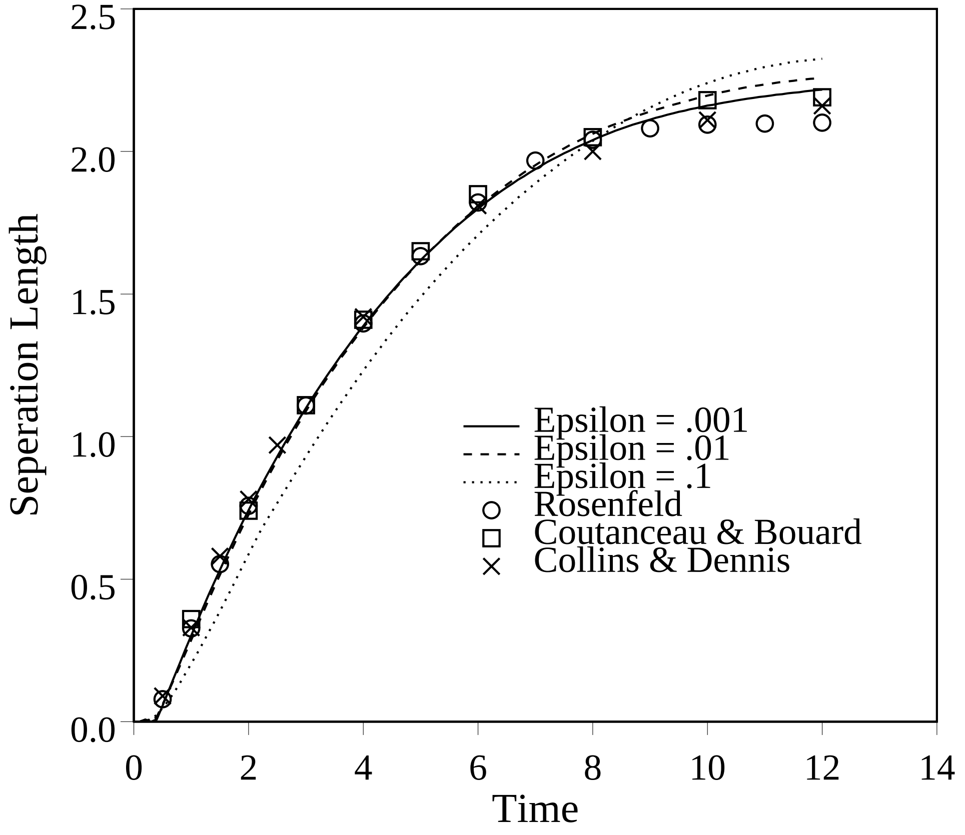


Fig. 6 Growth of separation bubble behind circular cylinder versus time.

The second problem computed for this geometry was the periodic vortex shedding at a Reynolds number of 100. In order to obtain the periodic shedding behavior in the least amount of time, the vortex shedding was triggered using an imposed tangential velocity over one half of the cylinder surface. The computations used a time step of 0.1, and were run for 400 iterations. For each case, Table 1 reports the CPU time in seconds on a Cray 2, the number of subiterations used, the lift and drag coefficients, and the Strohal number based on the frequency of the lift. These results show that the problem requires only that the maximum divergence of velocity be reduced below 10^{-3} to assure that enough subiterations are being used. However, the results from $\epsilon_c = 10^{-2}$ show that subiterating beyond this value will not change the solution very much, and that six to eight subiterations are adequate for this problem. These computed value of the Strouhal number (0.163) agrees well with many other reported results, including the experiment of Tritton[22] and of Kovasznay[23], and the computation of Braza et al.[24], who all reported values very near 0.16.

Table 1 Quantities for vortex-shedding calculations at Re = 100.

ϵ_c	Time	Subiterations	C_L	C_D	Strouhal No.
10^{-1}	370	2-5	± 0.424	1.393 ± 0.012	0.161
10^{-2}	820	6-8	± 0.362	1.377 ± 0.011	0.163
10^{-3}	1360	10-13	± 0.359	1.376 ± 0.011	0.163
10^{-4}	2610	17-26	± 0.358	1.376 ± 0.011	0.163

Artificial Heart Flow

The present flow solver is currently being used to compute the flow inside an artificial heart. The artificial heart was designed by Penn State University and is being studied experimentally by Tarbell *et al.*[25] The purpose of the current calculations is to demonstrate and analyze the present capability to compute a time-accurate incompressible flow through a complex internal device with moving-boundaries. The initial calculations in this effort are presented here. Since a number of primarily geometrical differences exist between these initial calculations and the actual artificial heart experiment, which are detailed below, only the simplest of qualitative comparison between computation and experiment can be made. Further work will attempt to remove these differences so that the actual artificial heart geometry can be accurately modeled.

The geometry used for the current model is depicted in Fig. 7. The heart is composed of a cylindrical chamber with two openings on the side for valves. The pumping action is provided by a piston surface which moves up and down inside the chamber. The diameter of the piston is 7.4 cm, with a stroke length of 2.54 cm. The problem was nondimensionalized with a unit length of 2.54 cm and a unit velocity of 40 cm/sec. The actual artificial heart has cylindrical tubes extending out of each of the side valve openings. These contain tilting flat disks which open and close to act as the valves. In the computational model these valves are not modeled, instead the boundary conditions at the side openings are specified to instantaneously open and close at the right moment. This simplification allows a single zone to be used to model the flow inside the chamber. With the addition of more zones, however, it will be possible to model the tilting disk valves as well. Additional simplifications made in the present computational model include the movement of the piston. In the actual device the piston moves through the entire chamber volume, including across the majority of the valve opening. Because of grid-generation problems, moving the piston past the valve openings would become quite difficult, and so the piston is restricted to move only in the volume beneath the valve openings. The flow is assumed laminar, and the Reynolds number based on the the unit length and velocity is set to 100. In the actual heart the Reynolds number is about 600, and regions of the flow are turbulent. Finally, the fluid is assumed to be Newtonian. This corresponds to the experiment of Tarbell *et al.*[25] who used a water and glycerin fluid whose viscosity is nearly the same as blood, about 3.5 centipoise, but unlike blood is simulated by a Newtonian fluid assumption.

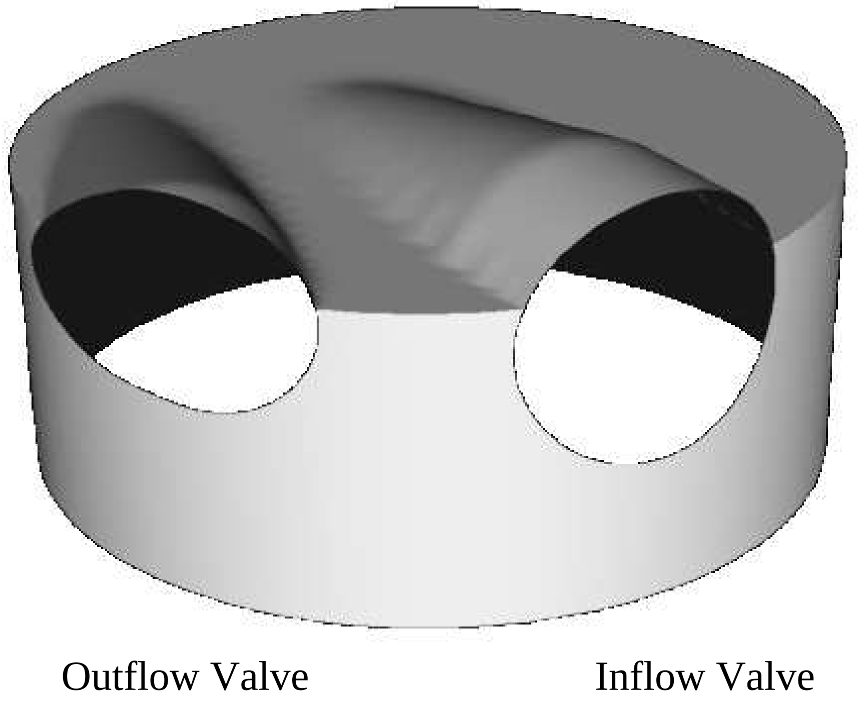


Fig. 7 Artificial-heart geometry showing valve openings.

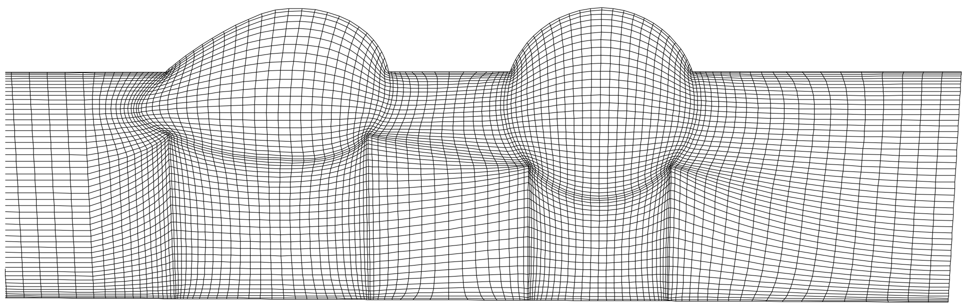


Fig. 8 Unwrapped grid used along side of the artificial heart near the region of the valve openings.

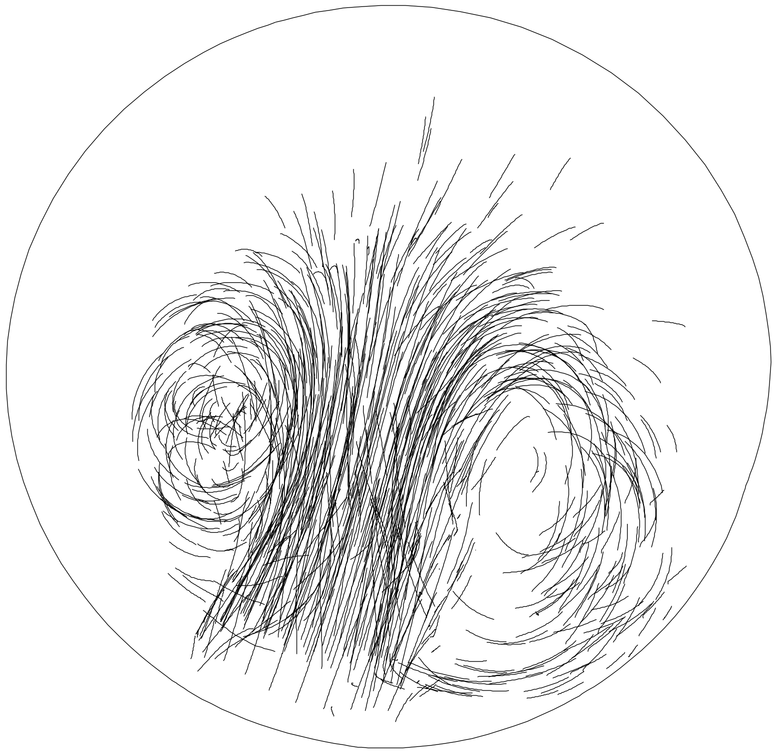
Inside the heart an H-H grid topology with dimensions of 39 x 39 x 51 is used. Figure 8 shows the grid on the unwrapped surface of the side of the heart chamber. Grid lines were placed along the lines of the valve openings to make implementation of the boundary conditions there straight forward. This surface grid was generated by dividing the surface into several zones and using a biharmonic grid generator[26] in each zone. The same grid generator was also used to generate an H grid on the piston surface and on the top surface. To fill in the interior points, an algebraic solver coupled with an elliptic smoother was used. As the piston moved up and down inside the chamber, the grid points below the valve openings were compressed and expanded, respectively. Thus a new grid was generated at each time step.

The flow was computed using a time step Δt of 0.025, and a β of 500. The piston moved with a constant nondimensionalized velocity of ± 0.2 between it's top and bottom positions, thus requiring 200 physical time steps for one period of the piston's motion. During each time step, the subiterations were carried out until the maximum residual converged dropped below 10^{-3} or until a maximum of 20 subiterations were used. This 20 subiteration limitation was used to restrict the overall computing time need for the computation. During most of the piston's cycle only 12-15 subiterations were required, but when the piston was changing directions, it did not completely converge in 20 subiterations. This "left over divergence" was quickly removed over the next few time steps, and did not greatly change the solution. The computing time required for each period of the piston's motion was approximately four hours. The computations were run for four periods during which time particle paths were computed after being released near the inflow valve.

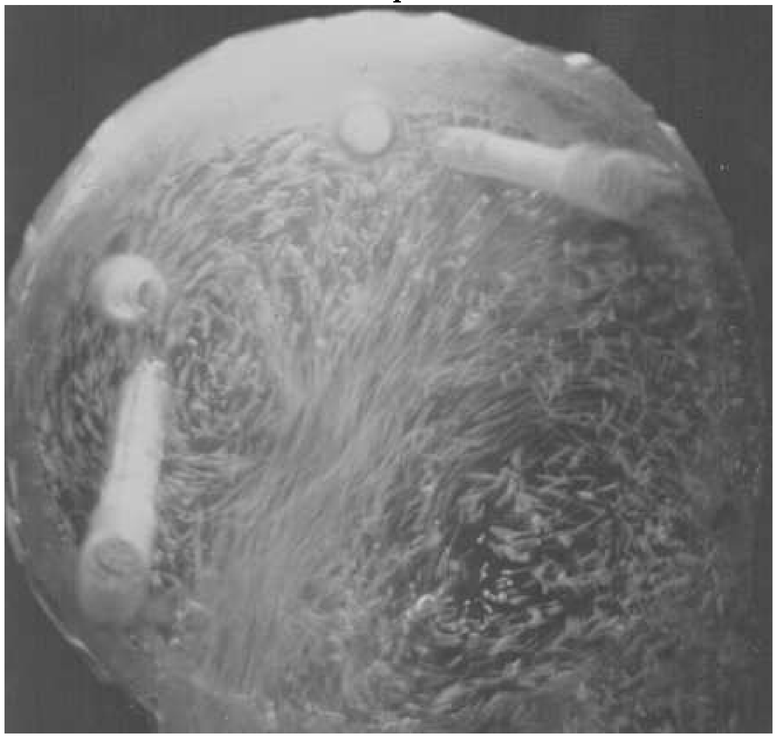
Figure 9a shows some of the computed particle traces as the piston nears it's bottom position. Two distinct vortices are seen to have formed from the flow separating as it enters through the inflow valve. Figure 9b shows an experimental photograph (J. M. Tarbell: private communication, 1988) of bubbles entering the inflow valve as the piston nears it's bottom position. A similar two-vortex system is seen to form here.

Computing Time

As expected, the computing time required for this code is greater than that required by a central difference, approximate factorization code, due to the complexity of the upwind-difference terms and the use of line-relaxation sweeps in more than one direction. The code runs between 100 to 200 microseconds per grid point per iteration on a Cray 2, depending on the number of sweeps taken. For two-dimensional problems the computing time required is between 20 to 35 microseconds per grid point per iteration on a Cray 2. The three-dimensional code uses 50 words of memory for each grid point. The advantage of this algorithm is its ability to converge quickly. For practical application, a steady-state solution is generally obtained after 4 orders of magnitude reduction of the maximum residual. This will be achieved in about 100 iterations, as was seen for the fine-grid duct case. Thus the computing time required for a steady-state solution using sweeps in all three directions is 0.02 seconds per grid point. As a point of comparison, the INS3D code[3], which uses central differencing and approximate factorization, runs at about 30 microseconds per grid point per iteration (using the diagonal algorithm version[27]) on a Cray 2. This code was used by McConnaughey et al.[28] to run the same duct flow problem as reported here. They reported using 2000 iterations to obtain a steady-state solution. Thus the INS3D code uses about 0.06 seconds per grid point per solution. The current code represents a factor of 20 increase in convergence rate, and a factor of 3 in overall computing time.



9a. Computation



9b. Experiment

Fig. 9 Incoming particle traces from computations and picture of experimental results as the piston nears the bottom position.

Another advantage to the present method is its robustness. When running a new problem the only numerical parameter which must be specified is the artificial compressibility parameter β. By contrast, the parameters which must be specified when running the INS3D code are β, the time step $\Delta\tau$, and the artificial dissipation coefficients. In addition, the range of acceptable values for β is significantly larger for the current algorithm, as illustrated by the duct problem. Thus the amount of computing time and the amount of user time spent finding a proper set of numerical values is greatly reduced in the present formulation.

For time-dependent flow problems, the current scheme will require on the order of 10 subiterations per time step. Thus the code will require on the order of 1000 microseconds per grid point per time step for three-dimensional problems. For problems in two-dimensions, the time is on the order of 200 microseconds per grid point per time step. This represents a significant amount of computational resources for most problems. This is on the order of the same amount of computing time required by a competing method, the fractional step as reported by Rosenfeld.[11] The problem primarily concerns the fact that the original equations are elliptic, thus the waves propagate information instantaneously throughout the domain. To do this computationally requires a lot computing time no matter what approach is taken.

Conclusion

An algorithm for computing both steady-state and time-accurate solutions to the incompressible Navier-Stokes equations has been presented. The method of artificial compressibility allows the equations to be solved as a hyperbolic system in pseudo-time. This requires the solution of a steady-state problem at each physical time step for the time-accurate formulation. The use of upwind differencing makes the system of numerical equations diagonally dominant. With the use of a nonfactored implicit line-relaxation scheme, the code can be run at very large time steps, and very fast convergence is seen. The results showed adequate comparison with experiment for the flow through a square duct. Comparison of flow over a circular cylinder for both an impulsively started flow and periodic vortex shedding show that when the maximum divergence of velocity is reduced to $10,^{-3}$ accurate integration in time is obtained. This required 10 to 13 subiterations per time step. Adequate solutions, however, could be obtained for this problem with even fewer subiterations. Finally, the computation of the flow through an artificial heart shows the capability of the code to simulate complicated internal flows with moving boundaries. Although the problems presented here are all for low Reynolds numbers, there is no inherent limit to the Reynolds number which can be computed with this method. The overall robustness of the code should help the code perform well at higher Reynolds numbers. Further advances in the convergence speed of the algorithm will still be very helpful in increasing the usefulness of this code as a design tool. This might be implemented using a multigrid cycle, and by studying ways to accelerate the convergence of the line-relaxation procedure.

Acknowledgments

The authors want to thank both Dr. Steve Ryan and Dr. Seokkwan Yoon for their helpful suggestions and discussions regarding the vectorization of the line-relaxation scheme. This work was partially sponsored by the NASA Technology Utilization Office.

References

[1] Rogers, S. E. and Kwak, D., "An Upwind Differencing Scheme For the Time-Accurate Incompressible Navier-Stokes Equations," AIAA Paper 88-2583, 1988, see also *AIAA J.*, Vol. 28, No. 2, 1990, pp. 253–262.

[2] Rogers, S. E. and Kwak, D., "An Upwind Differencing Scheme for the Steady-State Incompressible Navier-Stokes Equations," NASA TM 101051, 1988. Also submitted to *Appl. Num. Math.*

[3] Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarthy, S. R., "A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables," *AIAA J.*, Vol. 24, 1986, pp. 390–396.

[4] Merkle, C. L. and Athavale, M., "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," AIAA Paper 87-1137, 1987.

[5] Soh, W. Y. and Goodrich, J. W., "Unsteady Solution of Incompressible Navier-Stokes Equations," *J. Comp. Phys.*, Vol. 79, 1988, pp. 113–134.

[6] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comput. Phys.*, Vol. 43, p. 357, 1981.

[7] Chakravarthy, S. R., and Osher, S., "A New Class of High Accuracy TVD Schemes For Hyperbolic Conservation Laws," AIAA Paper 85-0363, 1985.

[8] Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations With Application to Finite Difference Methods," NASA TM 78605, 1979.

[9] Harten, A., Lax, P. D., and Van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *Siam Review*, Vol. 25, No. 1, p. 35, 1983.

[10] Harlow, F. H. and Welch, J. E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Physics of Fluids*, Vol. 8, 1965, pp. 2182–2189.

[11] Rosenfeld, M., Kwak, D., and Vinokur, M., "A Solution Method for the Unsteady Incompressible Navier-Stokes Equations in Generalized Coordinate Systems," AIAA Paper 88-0718, 1988.

[12] MacCormack, R. W., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, 1985.

[13] Chakravarthy, S. R., "Relaxation Methods for Unfactored Implicit Upwind Schemes," AIAA Paper 84-0165, 1984.

[14] Yoon, S., Kwak, D., and Chang, L., "LU-SGS Implicit Algorithm for Three-Dimensional

Incompressible Navier-Stokes Equations with Source Term," AIAA Paper 89-1964-CP, 1989.

15 Merkle, C. L. and Tsai, P. Y. L., "Application Of Runge-Kutta Schemes to Incompressible Flows," AIAA Paper 86-0553, 1986.

16 Chang, J. L. C. and Kwak, D., "On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows," AIAA Paper 84-0252, 1984.

17 Rogers, S. E., Kwak, D., and Kaul, U., "On the Accuracy of the Pseudocompressibility Method in Solving the Incompressible Navier-Stokes Equations," *Appl. Math. Modelling*, Vol. 11, 1987, pp. 35–44.

18 Humphrey, J. A. C., Taylor, A. M. K., and Whitelaw, J. H., "Laminar flow in a square duct of strong curvature," *J. Fluid Mech.*, Vol. 83, part 3, 1977, pp. 509–527.

19 White, F. M., Viscous Fluid Flow, McGraw-Hill, New York, p. 123, 1974.

20 Coutanceau, M. and Bouard R., "Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 2. Unsteady Flow." *J. Fluid Mech.*, Vol. 79, 1977, pp. 257–272.

21 Collins, W. M. and Dennis, S. C. R., "Flow Past An Impulsively Started Circular Cylinder," *J. Fluid Mech.*, Vol. 60, 1973, pp. 105–127.

22 Tritton, D. J., "Experiments on the Flow Past a Circular Cylinder at Low Reynolds Numbers," *J. Fluid Mech.*, Vol. 6, 1959, pp. 547–567.

23 Kovasznay, L. S. G., "Hot-Wire Investigation of the Wake Behind Cylinders at Low Reynolds Numbers," *Proc. Roy. Soc. A*, Vol. 198, 1949, pp. 174–190.

24 Braza, M., Chassaing, P., and Minh, H., "Numerical Study and Physical Analysis of the Pressure and Velocity Fields in the Near Wake of a Circular Cylinder," *J. Fluid Mech.*, Vol. 165, 1986, pp. 79–130.

25 Tarbell, J. M., Gunshinan, J. P., Geselowitz, D. B., Rosenburg, G., Shung, K. K., and Pierce, W. S., "Pulse Ultrasonic Doppler Velocity Measurements Inside a Left Ventricular Assist Device," *J. Biomech. Engr., Trans. ASME,* Vol. 108, 1986, pp. 232–238.

26 Bjorstad, P. E., "Numerical Solution of the Biharmonic Equation," Ph.D. Dissertation, Stanford University, 1980.

27 Rogers, S. E., Chang, J. L. C., and Kwak, D., "A Diagonal Algorithm for the Method of Pseudocompressibility," *J. Comp. Phys.*, Vol. 73, 1987, pp. 364–379.

28 McConnaughey, P., Cornelison, J., and Barker, L., "The Prediction of Secondary Flow in Curved Ducts of Square Cross-Section," AIAA Paper 89-0276, 1989.